

# Veritabanı Tasarımı

Büyük/Küçük Harf ve Karakter İşleme

## Konular

- Büyük/küçük harf dönüşümü ve karakter işleme yapan tek satır fonksiyonlarını uygulama
- SQL sorgularında büyük/küçük harf dönüşümü fonksiyonları: LOWER, UPPER ve INITCAP
- SQL sorgularında karakter işleme fonksiyonları: CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD, TRIM ve REPLACE
- Değişim değişkenleri kullanarak esnek sorgular yazmak

## Amaç

Hiç kendimizi hangi farklı şekillerde sunduğumuzu düşündünüz mü? Özel günler için farklı giyinirken, spor etkinlikleri için farklı konserler için farklı giyiniriz. Farklı durumlara nasıl baktığımız değişim için önemlidir. Kendinizi bir iş görüşmesi için nasıl göstermeyi seçersiniz?

Veri tabanındaki veri ile uğraşırken verilerin nasıl sunulduğu da önemlidir. SQL de çoğu zaman, isteklere bağlı olarak verinin görünümünü değiştirmemiz gerekir.

Bu derste verileri dönüştürmek için kullanılan çeşitli yolları öğreneceksiniz.

## DUAL Tablo

DUAL tablo 'X' ile ifade edilen bir satır ve 'DUMMY' ile ifade edilen bir sütuna sahiptir. Her hangi bir veritabanı tablosu ile direkt olarak ilişkili olmayan SELECT ifadelerinin oluşturulması ve fonksiyonların çalıştırılması için kullanılır. DUAL tablo kullanıldığında sonuç olarak sadece bir satır döner. Hesaplamalarda ve tablodan türetilmemiş ifadelerin değerlendirilmesinde kullanılır.

DUMMY
X

## DUAL Tablo (devam...)

DUAL bir çok tek-satır fonksiyonu öğrenmek için kullanılır. Bu örnekte DUAL tablosu hesaplama içeren SELECT ifadesi için çalıştırılmıştır. Gördüğünüz üzere SELECT ifadesi DUAL tablosunda var olmayan bir değer döndürmektedir. Geri dönen değer gerçekleştirilen hesaplamanın sonucudur.

```
SELECT (319/29) + 12  
FROM DUAL;
```

**(319/29)+12**

23

## Tek-satır Karakter Fonksiyonları

Tek-satır karakter fonksiyonları iki kategoriye ayrılır:

- Karakter katarlarını büyük/küçük harf olarak dönüştüren fonksiyonlar
- Karakter katarı fonksiyonları (join, extract, show, find, pad ve trim)

Tek-satır fonksiyonları SELECT, WHERE ve ORDER BY yantümceleri ile kullanılabilir.

## Tek-satır Karakter Fonksiyonları (devam...)

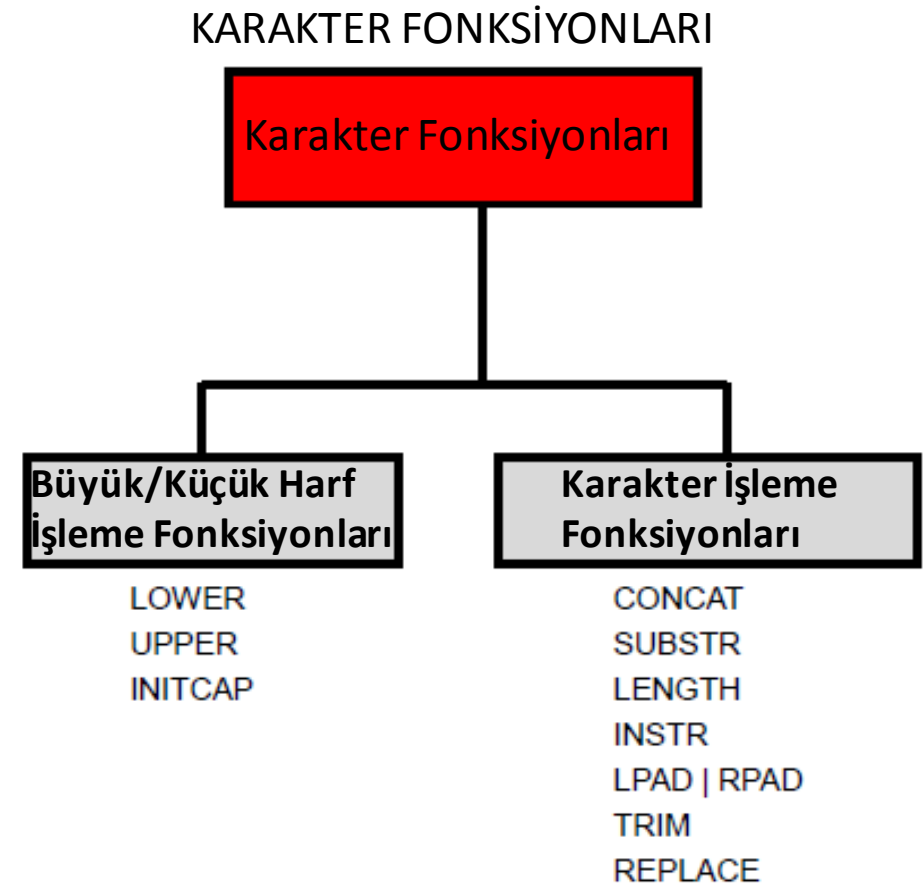
Büyük/küçük harf işleme fonksiyonları önemlidir çünkü veritabanında verilerin nasıl tutulduğunu(büyük, küçük ya da karışık olabilir) bilemeyiz.

Büyük/küçük harf işleme geçici olarak seçtiğiniz özelliğe göre veritabanınızdaki verileri dönüştürmenize izin verir.

## Büyük/Küçük Harf İşleme Fonksiyonları

Büyük/küçük harf işleme fonksiyonları büyük, küçük ya da karışık karakterlerin dönüştürülmesinde kullanılır. Bu dönüşümler çıkış ifadesini belli bir formata göre oluşturmak için kullanılırken özel bir karakter katarının aranmasında da kullanılır.

Büyük/küçük harf işleme fonksiyonları SQL ifadelerinin büyük çoğunluğunda kullanılır.





## Büyük/Küçük Harf İşleme Fonksiyonları (devam...)

Büyük/küçük harf işleme fonksiyonları aradığınız verinin büyük harf olarak mı küçük harf olarak mı tutulduğunu bilmediğiniz durumlarda yardımcı olur. Database tarafından bakıldığında 'v' ve 'V' aynı karakter değildir. Bu yüzden doğru şekilde arama yapmanız gerekir.

LOWER (sütun | ifade) alfa karakterleri küçük harfe dönüştürür.

```
SELECT title  
FROM d_cds  
WHERE LOWER(title) = 'carpe diem';
```

## Büyük/Küçük Harf İşleme Fonksiyonları (devam...)

UPPER (sütun | ifade) alfa karakterleri büyük harfe dönüştürür.

```
SELECT title  
FROM d_cds  
WHERE UPPER(title) = 'CARPE DIEM';
```

INITCAP (sütun | ifade) alfa karakterleri her kelimenin ilk harfi büyük olacak şekilde dönüştürür.

```
SELECT title  
FROM d_cds  
WHERE INITCAP(title) = 'Carpe Diem';
```

## Karakter İşleme Fonksiyonları

Karakter işleme fonksiyonları bir karakter dizesinin içinden seçim yapmak, değişiklik yapmak, biçimini değiştirmek için kullanılır.

Bir ya da daha çok karakter ya da kelimeler fonksiyonlara gönderilebilir ve fonksiyon verilen ifade üzerinde fonksiyonel özelliğini kullanarak değiştirilen ya da düzenlenen karakter katarını geri döndürür.

## Karakter İşleme Fonksiyonları (devam...)

**CONCAT:** İki değeri bir araya getirir.

**SUBSTR:** Karakter katarının istenilen uzunluğundaki bölümünü alır.

Fonksiyon	Sonuç
CONCAT('Hello', 'World')	HelloWorld
SUBSTR('HelloWorld', 1, 5)	Hello

## Karakter İşleme Fonksiyonları (devam...)

**LENGTH:** Karakter katarı uzunluğunu sayı değeri olarak verir.

**INSTR:** Verilen karakterin nümerik olarak yerini bulur.

**LPAD:** Karakterin sol tarafını doldurur, sağa yaslanmış değeri döndürür.

Fonksiyon	Sonuç
LENGTH('HelloWorld')	10
INSTR('HelloWorld','W')	6
LPAD(salary, 10,'*')	*****24000

## Karakter İşleme Fonksiyonları (devam...)

**RPAD:** Karakterin sağ tarafını doldurur, sola yaslanmış değeri döndürür.

**TRIM:** Başta ya da sonda bulunan belirtilmiş tüm karakterlerin kaldırılmasını sağlar. Trim fonksiyonu için sözdizimi şu şekildedir:

```
TRIM( [ leading | trailing | both  
[character(s) to be removed ] ] string to trim
```

Fonksiyon	Sonuç
RPAD(salary, 10, '*')	24000*****
TRIM('H' FROM 'HelloWorld')	elloWorld

## Karakter İşleme Fonksiyonları (devam...)

**REPLACE:** Karakter katarındaki belli bir karakter serisini başka bir karakter serisi ile yer değiştirir.

```
REPLACE (string1, string_to_replace, [replacement_string] )
```

- string1: değiştirilecek karakter katarı
- string\_to\_replace: aranan karakter serisi
- [replacement\_string]: yeni değiştirilecek karakter serisi

```
SELECT REPLACE('JACK and JUE','J','BL') "Changes"  
FROM DUAL;
```

## Fonksiyonlarla Sütun Takma Adı Kullanma

Tüm fonksiyonlar parantez içinde değerleri üzerinde çalışır. Ve her bir fonksiyon adı, sorgu oluştururken ki hatırlatmaya yardımcı amacını ifade eder.

Çoğu zaman bir sütun takma adı bir fonksiyon ismi olarak kullanılır. Bir sütun takma adı kullanıldığında, sütun takma adı gerçek fonksiyon sözdizimi yerine çıkışta görünür.



## Fonksiyonlarla Sütun Takma Adı Kullanma (devam...)

Gelecek örnekte takma ad "User Name" , ilk sorguda fonksiyon sözdizimi yerini aldı.

Sütun adındaki SELECT ifadesi sütun başlığı olarak görülür. Bununla birlikte ikinci sorgu örneğinde tabloda sonuç üretecek her hangi bir sütun yoktur. Bu yüzden sorgu sözdizimi kullanılır.

## Fonksiyonlarla Sütun Takma Adı Kullanma (devam...)

```
SELECT LOWER (last_name)||LOWER(SUBSTR(first_name,1,1))  
AS "User Name"  
FROM f_staffs;
```

User Name
does
millerb
tuttlem

```
SELECT LOWER (last_name)||LOWER(SUBSTR(first_name,1,1))  
FROM f_staffs;
```

LOWER (last_name)    LOWER(SUBSTR(first_name,1,1))
does
millerb
tuttlem

## Değişim Değişkenleri

Bazen farklı sonuç kümelerini almak için birçok farklı değerlerle aynı sorguyu çalıştırmanız gerekebilir. Eğer çalışanlar ve kendi bölümleri için bir rapor yazmak isterseniz, ancak sorgu sadece bir defada bir bölüm için veri döndürür.

Değişim değişkeni kullanımı olmadan, bu isteği defalarca SELECT ifadesini değiştirmek için aynı ifadeyi düzenlemek zorunda kalacağınız anlamına gelir.

## Değişim Değişkenleri (devam...)

Neyse ki, Oracle Application Express bizim için değişim değişkenlerini destekler. Bunları kullanmak için yapmanız gereken tek şey kodlanmış değeri **named\_variable:** ile değiştirmektir. Oracle Application Express daha sonra bu değeri ifadeyi çalıştırdığı zaman sizden isteyecektir.

## Değişim Değişkenleri (devam...)

Bu orijinal sorgu:

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id = 10 (and then 20, 30, 40...)
```

Şu şekilde yeniden yazılır:

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id = :dept_id
```

## Değişim Değişkenleri (devam...)

Değişim değişkenleri karakter veya tarih değerleri geçerken, Oracle Application Express tarafından karakter dizeleri olarak kabul edilir. Normalde karakter katarlarını çevrelemek için kullanacağınız tek tırnak işaretleri gerekmez.

WHERE yantümcesi şu şekilde görülür:

```
SELECT *  
FROM employees  
WHERE last_name = :l_name
```

## Değişim Değişkenleri (devam...)

Çalıştırma butonuna bastığınız zaman Oracle Application Express tarafından aşağıdaki gibi bir pop-up ekranda gözükür.

:L\_NAME

Submit